2Ш

УДК 001.51: 004.81

Е.А. Шестаков[1], А.А. Бутов[1], А.А. Воронов[1], Т.Л. Орлова[2]
[1]Объединенный институт проблем информатики НАН Беларуси, г. Минск, Беларусь
she@newman.bas-net.by, a.butov@sam-solutions.net, sash_v_oo@tut.by
[2]«КБТЭМ-ОМО» ГНПО «Планар», г. Минск, Беларусь, tlo@list.ru

Автоматизированная система подготовки информации для изготовления фотошаблонов

Предлагается описание автоматизированной системы, посредством которой по описанию топологии микроэлектронных схем подготавливается входная информация для устройств, выполняющих изготовление фотошаблонов. Входная информация представляет собой описание последовательности прямоугольников. Выбор этой последовательности во многом определяет производительность этих устройств и качество получаемых фотошаблонов. Приводится краткое описание задач, решаемых системой, а также ее структура.

Введение

При производстве интегральных схем, фотоэлектрических преобразователей, ЖК-индикаторов, а также многих других микроэлектронных устройств возникает задача формирования топологических структур на металлизированных фотошаблонах [1]. Эти структуры формируются с помощью специальных генераторов изображений. Генераторы изображений строят топологию на фотошаблоне из наборных элементов. Наборный элемент представляет собой прямоугольник. Создание посредством таких генераторов произвольных изображений топологических структур требует предварительного разложения (декомпозиции) описания этих структур на множество прямоугольников, объединение которых с заданной точностью совпадает с описанием соответствующих исходных структур. При этом число прямоугольников, входящих во множество, должно быть минимальным или близким к минимальному. По полученному множеству прямоугольников формируется оптимальная входная последовательность, которая кодируются в соответствии с правилами входного языка соответствующего генератора изображений. Существует целый ряд автоматизированных систем [1], [3-6] подготовки входной информации для генераторов изображений, однако они либо ориентированы на ограниченный класс топологических объектов, либо недоступны.

В настоящей работе предлагается описание автоматизированной системы подготовки входной информации для одноканальных генераторов изображений. Система дает возможность формировать по описанию топологии, представленному или в виде файла в GDS II [2] формате, или в виде текстового файла (внутренний формат) входной файл для генератора изображений.

Задачи, решаемые системой

Формирование входной информации для генераторов изображений выполняется в несколько этапов.

Этап 1. Если описание топологии фотошаблона представлено в виде файла в GDS II формате [2], то выполняется преобразование этого описания в описание

топологии в виде множества базовых топологических объектов типа контур, окружность и шина, представленных в виде текстового файла. Каждый базовый объект в этом файле задается одной строкой. Для контура в строке указывается число вершин и координаты вершин контура. Шина представляется шириной шины и ломаной, задающей трассу шины. Окружность задается координатами центра, длиной радиуса и числом, определяющим точность аппроксимации окружности контуром. Исходное описание топологии фотошаблона также может быть задано текстовым файлом в виде множества контуров, шин и окружностей.

Этап 2. Выполняется анализ множества контуров на корректность их задания. Контур считается некорректно заданным, если длина некоторых его сторон равна нулю или меньше минимальной длины стороны прямоугольника наборного элемента. Также невозможна ситуация, когда угол между соседними сторонами контура равен нулю. Некорректно заданные контуры удаляются из множеств контуров, описывающих топологию фотошаблона.

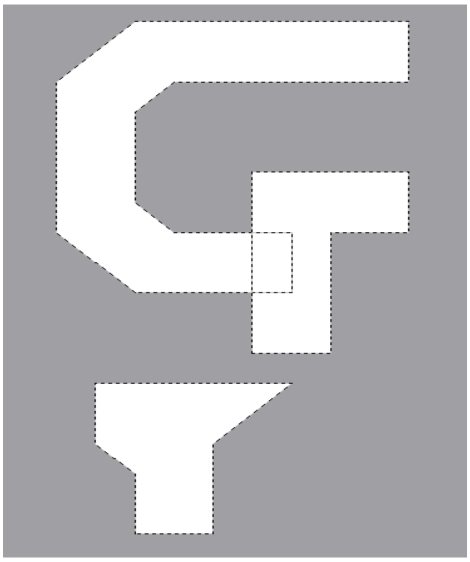


Рисунок 1 – Отображение входных данных

Этап 3. Осуществляется поиск пересекающихся между собой топологических объектов, заданных контурами. Если пересекающиеся топологические объекты существуют, то исходное множество этих объектов разбивается на классы. Топологические объекты a и b принадлежат одному классу, если в этом классе имеются топологические объекты c_1, c_2, …, c_n, такие, что пары объектов ac_1, c_1c_2, c_2c_3, …, $c_{n-1}c_n$, c_nb пересекаются между собой, т.е. пересечение этих объектов не равно пустому множеству.

Пример выполнения этапа 3 представлен на рис. 1. На этом рисунке представлены три топологических объекта. Пересечение двух из них не является пустым. Следовательно эти объекты разбиваются на два класса.

Этап 4. Если на множестве топологических объектов задано разбиение, то выполняется объединение топологических объектов, представленных контурами, входящими в один класс, т.е. множество контуров, входящих в один класс заменяется одним контуром. После выполнения этого этапа описание исходной топологии представляется в виде множества не пересекающихся между собой объектов топологии.

Этап 5. Из множества топологических объектов, полученных в результате выполнения 4-го этапа, выделяются односвязные и многосвязные топологические объекты (контуры). Односвязный топологический объект не содержит внутри себя каких-либо других топологических объектов. Многосвязный топологический объект представляется рядом контуров. Первый контур этого ряда является основным. Он содержит внутри себя все остальные контуры этого ряда, называемые контурами-разрезами. Контур-разрез не может находиться внутри какого-либо другого контура кроме основного. Точки плоскости, находящиеся внутри основного контура или на границе основного контура или контуров-разрезов, но не внутри контуров-разрезов, являются точками многосвязного топологического объекта. Односвязный топологический объект рассматривается как частный случай многосвязного объекта. При этом единственный контур, задающий его, является основным.

Результат выполнения этого этапа представляется в виде текстового файла, каждая строка которого задает либо односвязный, либо многосвязный контур. Многосвязный контур задается последовательностью контуров, первый из которых является основным.

Для объектов, представленных на рис. 1, в результате выполнения этапа 5, получаем два односвязных контура (рис. 2).

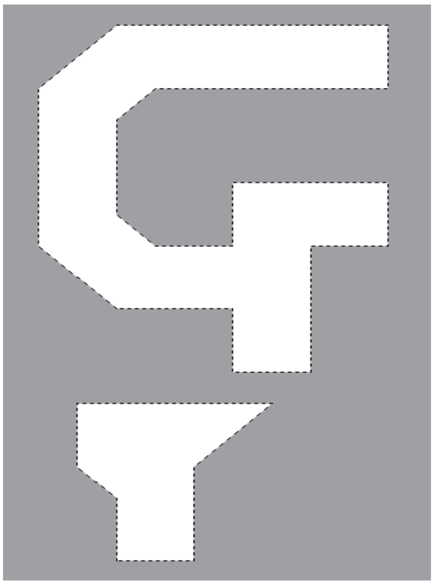


Рисунок 2 – Отображение полученных данных на этапе 5

Этап 6. Некоторое подмножество односвязных топологических объектов, заданных контурами, входящими в заданный прямоугольник, может быть инвертировано. При этом предполагается, что в заданный прямоугольник не входят многосвязные топологические объекты. В результате выполнения этого этапа рассматриваемое подмножество односвязных топологических объектов (контуров) заменяется одним многосвязным топологическим объектом, в котором в качестве основного контура выступает заданный прямоугольник, а контуры, задающие односвязные топологические объекты данного подмножества, являются контурами-разрезами.

Этап 7. Для каждого односвязного и многосвязного топологического объекта решается задача декомпозиции (покрытия), то есть представления каждого из этих

объектов в виде совокупности прямоугольников, объединение которых с заданной точностью совпадает с исходным объектом. Точность аппроксимации определяется ограничением на минимальную длину сторон прямоугольников покрытия.

Результат выполнения этого этапа для рассматриваемого примера представлен на рис. 3.

Этап 8. По полученному множеству прямоугольников, покрывающих исходные топологические объекты, формируется оптимальная последовательность прямоугольников, которая кодируются в соответствии с правилами входного языка соответствующего генератора изображений.

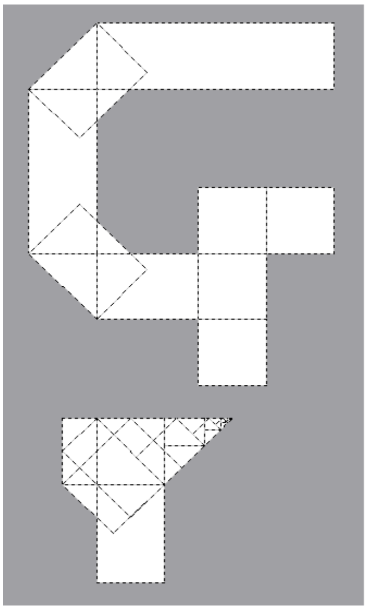


Рисунок 3 – Результат представления данных для лазерного генератора изображений

Метод декомпозиции

Для решения задачи 7-го этапа используются как специальные, так и универсальные алгоритмы. Специальные алгоритмы решают задачу покрытия окружностей, колец, шин, треугольников. Универсальный алгоритм решает задачу покрытия произвольного многоугольника, заданного одним или несколькими контурами.

Универсальный алгоритм декомпозиции, применяемый в системе, является эвристическим и состоит из следующих этапов.

1. Множество всех сторон исходного многосвязного топологического объекта разбивается на простые классы. В один простой класс включаются стороны параллельные или перпендикулярные друг другу. Для каждого простого класса полученного разбиения вычисляется вес, равный сумме длин сторон, входящих в него. Простые классы разбиения упорядочиваются по убыванию весов. Стороны внутри каждого простого класса упорядочиваются по убыванию своих длин.

2. Для каждой вершины основного контура, а также контура-разреза определяется ее тип. Тип вершины определяется по внутреннему углу, образованному двумя сторонами контура, примыкающими к данной вершине. Рассматриваются четыре типа вершин, которые помечаются числами от 0 до 3. Считается, что внутренний угол между двумя сторонами, примыкающими к данной вершине, не может равняться нулю и 360°. Поэтому если этот угол меньше 90°, то вершина относится к

типу 3. Если внутренний угол между сторонами больше или равен 90°, но меньше 180°, то вершина многоугольника относится к типу 2. Если угол между сторонами больше или равен 180°, но меньше 270°, то вершина многоугольника относится к типу 0. Если угол между сторонами больше или равен 270°, но меньше 360°, то вершина многоугольника относится к типу 1.

Внутренние углы контуров-разрезов являются внешними для многосвязного многоугольника. Поэтому для контуров-разрезов проводится корректировка типов вершин.

Всякая сторона многосвязного топологического объекта характеризуется парой чисел, задающих тип вершин, на которые она опирается. По этому признаку все стороны многоугольника разбиваются на 16 классов типов: (3, 0), (3, 1), (3, 2), (3, 3), (2, 0), (2, 1), (2, 2), (2, 3), (1, 0), (1, 1), (1, 2), (1, 3), (0, 0), (0, 1), (0, 2), (0, 3).

3. По каждой стороне топологического объекта, относящейся к типу из подмножества {(2, 0), (2, 1), (2, 2), (1, 0), (1, 1), (1, 2), (0, 0), (0, 1), (0, 2)}, строится один прямоугольник искомого покрытия. Одна из сторон этого прямоугольника содержит рассматриваемую сторону топологического объекта. Этот прямоугольник имеет максимально возможную площадь.

4. По каждой стороне топологического объекта, относящейся к типу из подмножества {(3, 0), (3, 1), (3, 2), (3, 3), (2, 3), (1, 3), (0, 3)}, строится несколько прямоугольников искомого покрытия. Одна из сторон этих прямоугольников лежит на рассматриваемой стороне или содержит часть этой стороны. Хотя бы один из внутренних углов, прилегающих к данной стороне, является острым. Для такого угла находится ряд прямоугольников покрытия, объединение которых содержит максимально возможную область исходного объекта, расположенную между этой стороной и биссектрисой острого угла.

Этот алгоритм в ряде случаев не гарантирует, что полученное подмножество прямоугольников покрывает данный топологический объект. Поэтому после получения совокупности прямоугольников (покрытия) решается задача анализа этой совокупности на корректность. Для этого прямоугольники полученной совокупности объединяются в один односвязный или многосвязный топологический объект. Если данная совокупность прямоугольников является покрытием многосвязного объекта, то в это объединение добавляются и объекты, задаваемые контурами-разрезами. Если в результате объединения прямоугольников получается односвязный объект, то полученное покрытие является корректным. Если результирующий объект является многосвязным, то покрытие не является корректным. Некорректное покрытие дополняется прямоугольниками так, чтобы оно стало корректным. Для этого для каждого контура-разреза многосвязного топологического объекта, полученного в результате объединения, находится множество прямоугольников, содержащих этот контур-разрез. Полученные таким образом множества прямоугольников добавляются в покрытие.

Структура системы

Задачи, решаемые в рамках описанных выше этапов, реализованы в виде набора соответствующих подпрограмм, которые интегрированы в специально разработанную графическую систему визуального отображения **Polygon**. Эта система существенно облегчает работу проектировщика, позволяя выполнить комплексное (сквозное) топологическое проектирование с просмотром и анализом промежуточных результатов, а также предоставляет целый ряд сервисов, упрощающих процесс проектирования.

Система **Polygon** реализует многодокументный интерфейс и соответственно состоит из главного окна, снабженного меню и набором плавающих инструментальных панелей, и дочерних окон, открываемых по мере необходимости. Последние, собственно, и отображают в графической форме всю информацию, связанную как с исходными топологическими объектами, так и результатами, получаемыми на тех или иных этапах проектирования. Типичный вид системы **Polygon** с тремя открытыми дочерними окнами изображен на рис. 4.

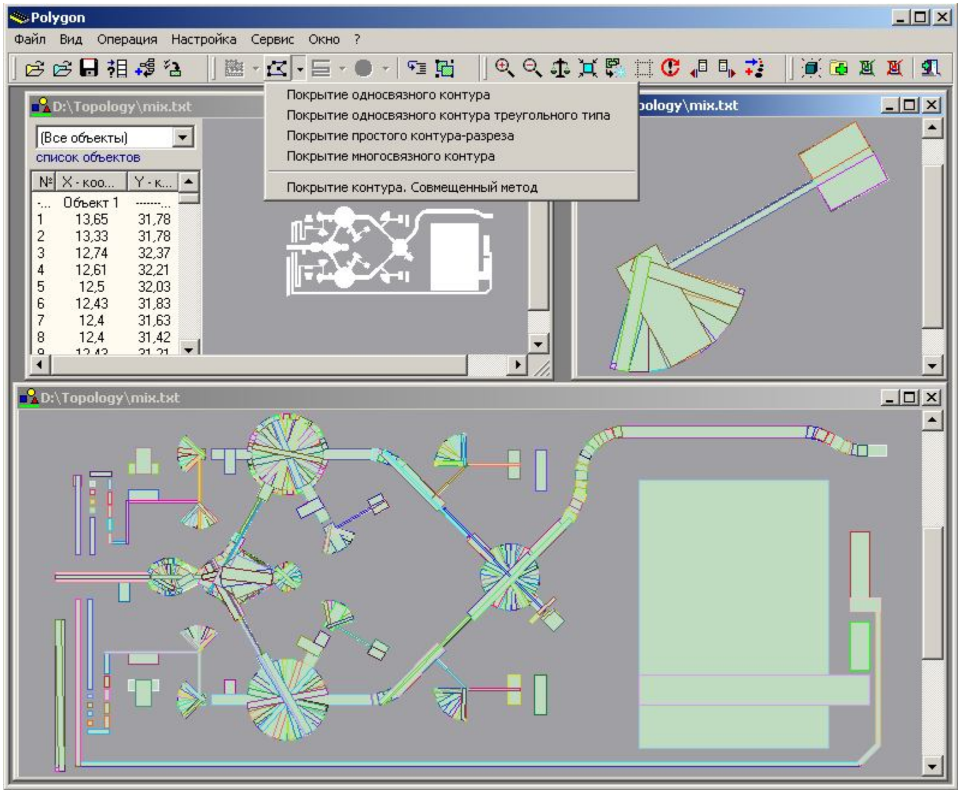


Рисунок 4 – Типичный вид системы **Polygon**

Основными функциями, выполняемыми системой, являются следующие.

1. Преобразование топологической информации из общепринятого формата представления GDS II во внутренний формат.

2. Выбор и открытие файлов на диске, содержащих исходную или промежуточную топологическую информацию. Чаще всего открытие такого файла влечет за собой открытие соответствующего дочернего окна, в котором отображается один или несколько топологических объектов. При этом сразу же можно получить информацию о типах топологических объектов (контур, шина, круг, кольцо), их числе, а также о том, какие из имеющихся контуров являются четырехугольниками (но не прямоугольниками), а какие – прямоугольниками. Эта информация фиксируется в протоколе сеанса работы системы.

3. Обнаружение и устранение аномалий в исходных данных. При наличии аномалий на экран выводится соответствующее сообщение, а детальная информация заносится в протокол сеанса работы системы. Далее пользователь имеет возможность отменить или продолжить начатое действие. При выборе варианта продолжения все строки с аномальными данными будут скорректированы или удалены из исходных данных.

4. Протоколирование специфических показателей и важнейших параметров работы выполняемых подпрограмм с целью их последующего анализа. Автоматически создаваемый протокол сеанса работы системы можно записать в файл, либо дописать в конец существующего файла протокола.

5. Запуск подпрограмм, реализующих те или иные этапы топологического проектирования применительно к одному или нескольким топологическим объектам, содержащимся в активном дочернем окне. При наличии альтернативных алгоритмов делается выбор одного из них. Если решается задача нахождения покрытия, то каждый элемент, вошедший в покрытие, подвергается проверке: является ли он прямоугольником или нет. Результат проверки записывается в протокол сеанса работы системы.

6. Сохранение в файле на диске полученной промежуточной или результирующей топологической информации. Имена файлов автоматически снабжаются соответствующими окончаниями, отражающими специфику содержащейся информации.

7. Использование пошагового режима вывода на экран элементов найденного покрытия. В этом случае прямоугольники, образующие покрытие, будут показаны в дочернем окне не все сразу, как это реализовано по умолчанию, а поочередно в той последовательности, в какой они были получены алгоритмом поиска покрытия. Пользователь сам управляет стартом каждого шага и может в любой момент прервать процесс последовательного показа элементов покрытия.

8. Оперативный анализ текущей информации, отображаемой в активном дочернем окне. При необходимости можно многократно увеличить или уменьшить масштаб изображения, убрать или уменьшить координатную панель, увеличивая тем самым клиентскую область окна с целью более детального рассмотрения графической фигуры или ее фрагмента. Предусмотрены операции, связанные с центрированием изображения после изменения размеров окна, восстановлением исходного размера фигуры после многократного выполнения операций масштабирования изображения, обновлением изображения, показом и удалением найденного покрытия и др. Выбор на координатной панели конкретной строки (пары значений координат X и Y) приводит к графическому выделению соответствующей граничной точки на изображении объекта с помощью кружка-метки.

При решении задач покрытия и отображении элементов покрытия предусмотрена возможность прорисовки границ исходных топологических объектов пунктирной линией для выяснения, не захватывает ли тот или иной элемент покрытия часть пространства за пределами контура покрываемого объекта.

9. Многооконный режим выполнения топологического проектирования. Новый набор топологических объектов можно отобразить либо в уже открытом активном дочернем окне с заменой прежнего изображения, либо в новом дочернем окне. Можно открыть несколько дочерних окон, отображая в них либо различные фрагменты топологической структуры, либо части одного и того же фрагмента в различных масштабах, либо результаты работы альтернативных алгоритмов над одними и теми же исходными данными, либо результаты последовательного выполнения различных этапов проектирования.

Новые возможности системы связаны с использованием так называемых проектов, каждый из которых позволяет группировать и хранить в базе данных исходные, промежуточные и результирующие данные, относящиеся к конкретной топологической структуре (объекту, набору объектов или слою топологии СБИС) как единое целое. При открытии конкретного проекта можно, в частности, увидеть, какие этапы проектирования уже были выполнены, а также хронологию их выполне-

ния во времени. Если проектирование еще не закончено, его можно продолжить. При необходимости тот или иной уже выполненный этап проектирования можно повторить заново, используя, например, альтернативные или вновь разработанные методы решения, доступные в обновленной версии системы. В этом случае ранее полученные данные, связанные с последующими этапами проектирования, удаляются, и эти этапы должны быть выполнены заново с использованием обновленной входной информации. Другими словами, хронология выполнения этапов проектирования всегда должна соответствовать формальному порядку их следования в используемой технологии топологического проектирования.

Заключение

Предлагаемая автоматизированная система дает возможность контролировать все этапы формирования входной последовательности для генераторов изображений. Для некоторых из этих этапов предлагаются альтернативные варианты их решения. Все это дает возможность формировать более качественные входные последовательности для генераторов изображений.

Литература

1. Фейнберг В.З. Геометрические задачи машинной графики больших интегральных схем. – М.: Радио и связь 1987. – 178 с.
2. GDS II format. – Режим доступа: http://www.xs4all.nl/~kholwerd/interface/bnf/gdsformat.html
3. Осипов Л.Б., Широ Г.Э. Алгоритмические методы подготовки и контроля информации для микрофотонаборных установок // Электронная техника. – Сер. 10. Микроэлектронные устройства. – 1978. – Вып. 2. – С. 89-103.
4. Широ Г.Э., Осипов Л.Б., Волков В.И. Параллельный быстродействующий алгоритм подготовки для микрофотонаборных установок // Электронная техника. – Сер. 11. Комплексная микроминиатюризация радиоэлектронных устройств и систем. – 1976. – Вып. 3. – С. 71-76.
5. Nahar S.and Sahni S., A Fast Algorithm for Polygon Decomposition, // IEEE Trans. on 38 Computer Aided Design of Integrated Circuits and Systems. – Vol. CAD-7, No. 4. – April 1988. – P. 478-483.
6. Berman P. and Dasgupta B. Approximating Rectilinear Polygon Cover Problems. Algorithmica. – 17(4). – 1997. – P. 331-356.

E.A. Shestakov, A.A. Butov, A.A. Voronov, T.L. Orlova

The Automated System of Data Preparation for Making Photomasks

We present the automated system of data preparation for integrated circuit layout generators. We consider the problem of covering polygons with rectangles. Input data is the sequence of rectangles. Basically this sequence determines the productivity of these integrated circuit layout generators and quality of output photomask. We describe the tasks which are solved by system, functions and structure of the system and give some examples.

Статья поступила в редакцию 17.07.2008.